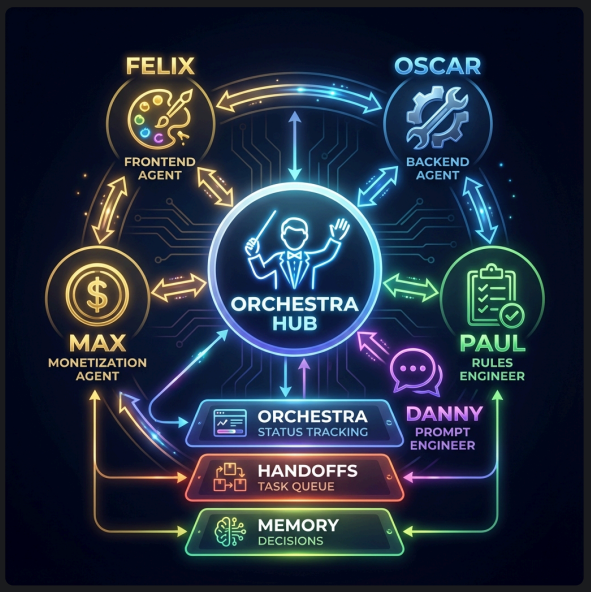
Multi-Agent Orchestration System

> *A comprehensive guide to how specialized AI agents collaborate on your projects.*

Overview

Your workspace uses an **orchestrated multi-agent system** where specialized AI agents work together on complex projects. Each agent has distinct expertise and can hand off work to others.



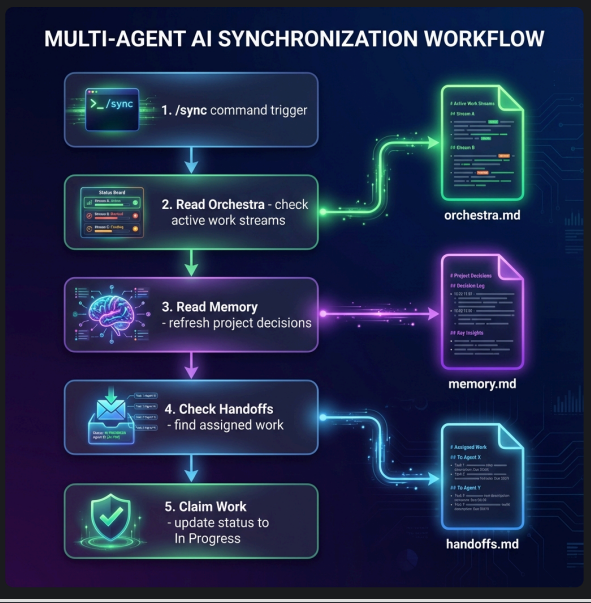
The Agent Team

Agent	Role	Specialization	Model
🎨 **Felix**	Frontend	UI/UX, CSS, animations, themes, accessibility	Gemini
🔧 **Oscar**	Backend	APIs, databases, Supabase, authentication, security	Claude
📋 **Paul**	Rules Engineer	Workflows, system rules, issue tracking, best practices	Any
💬 **Danny**	Prompt Engineer	AI prompts, structured outputs, CRAFT framework	Claude Opus
💰 **Max**	Monetization	Payments, pricing, analytics, conversion optimization	Any

Invoking agents: Use slash commands `/felix`, `/oscar`, `/paul`, `/danny`, `/max`

When summoned, agents: (1) Run `/sync` to check work, (2) Review project state, (3) Pick up assigned work or ask what's needed.

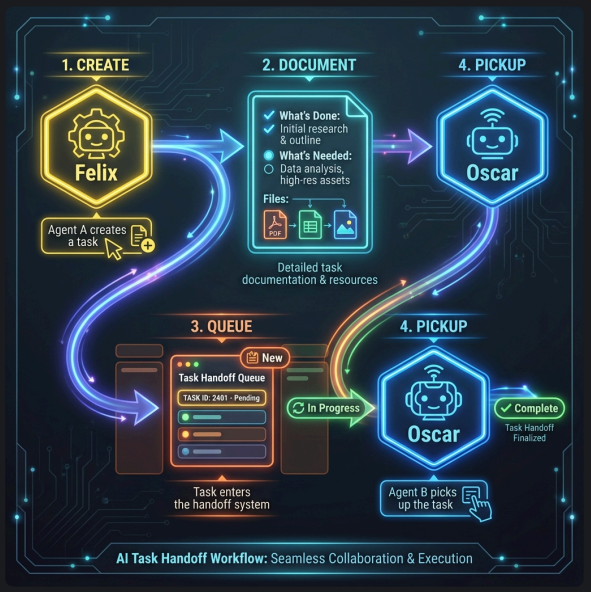
Shared Context System


MULTI-AGENT AI SYNCHRONIZATION WORKFLOW
1. /sync command trigger
2. Read Orchestra - check active work streams
3. Read Memory - refresh project decisions
4. Check Handoffs - find assigned work
5. Claim Work - update status to In Progress
orchestra.md
memory.md
handoffs.md

File	Purpose
orchestra.md	Central coordination — tracks active streams, status, completed work
handoffs.md	Task queue — work items being passed between agents
memory.md	Persistent memory — architecture decisions, tech stack, gotchas

---

The Handoff Protocol



When an agent completes work, they perform a **handoff**:

Step	Action
1. Document	Commit code, document new files, ensure working state
2. Create Prompt	Write detailed prompt: what's done, what's needed, files
3. Queue	Add entry to handoffs.md with 🆕 New status
4. Pickup	Receiving agent runs `/sync`, claims as 🔄 In Progress

Key Workflows

`/sync` - Context Synchronization

Read Orchestra → Read Memory → Check Handoffs → Read Issues Log → Claim work

`/orchestrate` - Multi-Agent Coordination

Read orchestra state → Update your stream → Check handoffs → Claim work → Update memory

`/handoff` - Passing Work

Document completed work → Create handoff entry → Update orchestra → Notify user

---

Status Legend

Status	Meaning
🟢 **Active**	Currently working
🟡 **Waiting**	Blocked or awaiting handoff
🔵 **Review**	Needs user review
✅ **Done**	Completed
🆕 **New**	Just added, not picked up
🔄 **In Progress**	Being worked on

Summary

Capability	How It Works
Specialization	Each agent focuses on their expertise
Continuity	Shared context persists across sessions
Coordination	Orchestra tracks who's doing what
Handoffs	Work flows smoothly between agents
Memory	Decisions are recorded for future reference